CCNA v1.1 Study Guide

Alexander

June 18, 2024

information compiled from:

- *CCNA 200-301 Volume 1&2 Official Cert Guide: Advance your IT career with hands-on learning* by Wendell Odom
- https://www.youtube.com/@JeremysITLab
- https://study-ccna.com/
- https://chatgpt.com/

1. Network Fundamentals (20%)
2. Network Access (20%)
3. IP Connectivity (25%)
4. IP Services (10%)
5. Security Fundamentals (15%)
6. Automation and Programmability (10%)

Network Fundamentals

- **Explain the role and function of network components**

  Routers forward packets between computer networks (LANs and WANs), and use IP addresses to determine where a packet should go. Routers remove data-link headers and trailers, analyze packet information, and then re-encapsulate the data with a new data-link frame before transmission. Routers compare destination IPs, found inside packet headers, to routes in routing tables before forwarding packets. Routing tables get their routes from static routes, dynamic routing protocols, and source headers fields found in packets recieved on local interfaces. Routers can perform many different actions: implementing QoS tools, permiting/denying packets with ACLs, performing firewall functions, advertising BSSs, and more.

Switches forward PDUs at layer 2 within the same network segment. Switches place new source MAC addresses, found in the header fields, into a table. This MAC-address table helps the switch make forwarding decisions. Switch ports divide collision domains; unlike router ports, which divide broadcast domains. Switches can divide broadcast domains by using VLANs, but cannot route between subnets. Switches can perform a variety of security tasks as well like: port security, DAI, DHCP snooping. L3 switches can perform both switching and routing functions.

Firewalls sit in the path that packets take through the network. They permit/deny traffic much like an ACL would do on a router. Firewalls are capable of watching application-layer flows with AVC, performing webpage verification on URIs, and retaining state. IPSs can compare packet flows to exploit signatures, log events, and can discard/redirct packets.

An AP serves as a single point of contact for every device that wants to use a BSS. An AP uses a unique BSSID that is based on the AP's own radio MAC address. APs are often connected to a DS that uplinks to a wired network.

The purpose of a controller, within a network, is to centralize control plane functions. It is primarily focused on controlling the flow of data within the network, making rouing decisions, enforcing policies for security or QoS, and dynamically configuring network devices. Controllers typically interact with a Network Management System (NMS) via a NorthBound API or NorthBound Interface (NBI), and can interact with the network devices via a SouthBound API (SBI).

A WLC manages lightweight APs. They are responsible for optimizing RF utilization, authenticating clients (802.1X), encrypting data (WPA2/3), monitoring, reporting and more.

WLC ports are physical networking interfaces that allow the controller to connect to other network devices. There are a varity of ports types:

Service ports are used for out-of-band mangement, system recovery, and initial boot functions; always connects to a switch port in access mode. These ports will normally be in a management VLAN.

Console ports are used for out-of-band management, system recovery, and initial boot functions; asynchronous connection to a terminal emulator (9600 baud, 8 data bits, 1 stop bit)

Redundancy ports are used to connect to a peer controller for high availability (HA) operation.

Distribution system ports carry most of the data coming to and going from the controller. These normally connect using 802.1Q trunk mode. These ports should be configured as an unconditional EtherChannel, since Cisco WLCs do not support LACP/PAgP. Many different types of traffic flow via this channel: CAPWAP tunnel traffic, client data pass from wireless LANs to wired VLANs, plus any mangement traffic (HTTPS, SSH, SNMP, TFTP, etc.). Because the DS ports must carry data that is associated with many different VLANs, VLAN tags and numbers become crucial. The DS ports can operate independently, each one transporting multiple VLANs to a unique group of internal controller interfaces.

WLC ports are physical networking interfaces that allow the controller to connect to other network devices. There are a varity of ports types:
Service ports are used for out-of-band mangement, system recovery, and initial boot functions; always connects to a switch port in access mode. These ports will normally be in a management VLAN.
Console ports are used for out-of-band management, system recovery, and initial boot functions; asynchronous connection to a terminal emulator (9600 baud, 8 data bits, 1 stop bit)
Redundancy ports are used to connect to a peer controller for high availability (HA) operation. Distribution system ports carry most of the data coming to and going from the controller. These normally connect using 802.1Q trunk mode. These ports should be configured as an unconditional EtherChannel, since Cisco WLCs do not support LACP/PAgP. If you want you could set the DS ports up to operate independently, each one transporting multiple VLANs to a unique group of internal controller interfaces. For resiliency, you can configure DS ports in redundant pairs. One port is primarily used; if it fails, a backup port is used instead. An Etherchannel is probably ideal in most cases. It is important to note that many different types of traffic flow via this channel: CAPWAP tunnel traffic, client data passing from the wireless LANs to wired VLANs, plus any mangement traffic (HTTPS, SSH, SNMP, TFTP, etc.). Because the DS ports must carry data that is associated with many different VLANs, VLAN tags and numbers become crucial.

WLC interfaces are logical network connections configured within the controller's OS. Cisco controllers support the following interface types:
Management interfaces are used for normal management traffic, such as RADIUS user authentication, WLC-to-WLC communication, web-based and SSH sessions, SNMP, NTP, syslog, and so on. The management interface is also used to terminate CAPWAP tunnels betweeen the controller and its APs.

Redundancy management interfaces hold the management IP address of a redundant WLC that is part of a HA pair of controllers. The active WLC uses the management interface address, while the standby WLC uses the redundancy management address.

Virtual interfaces hold the IP address facing wireless clients when the controller is relaying client DHCP requests, performing client web authentication, and supporting client mobility.

Service port interface is bound to the service port and used for out-of-band management.

Dynamic interfaces connect a VLAN to a WLAN.

Endpoints are devices or nodes within a network that communicate directly with users, applications, or other devices. They serve various purposes depending on their type and functions. An endpoint could be a server, laptop, IoT devices, cameras, VoIP phones. Endpoints facilitate user interaction, consume and create data, enable communication, process data, or can aid with security of an organization.

Servers are fundamental to network operations, providing centralized resources and services to clients or other devices within a network. They perform various crucual functions: data storage/retrieval, application hosting, resource sharing, security and authentication, backup and recovery.

Power over Ethernet (PoE) technology enables the transmission of electrical power alongside data over standard Ethernet cables. It plays a crucial role in network deployments where you have an endpoint that needs power but no infrastructure to support it. PoE standards can vary, but 802.3bt can support up to 100 watts!

- **Describe characteristics of network topology architectures**

  In a two-tier design (collapsed core) you have an access and distribution layer. Access Switches connect directly to end users, providing user devices access to the LAN. Access switches normally send traffic to and from the end-user devices to which they are connected and sit at the edge of the LAN. Distribution switches provide a path through which the access switches can forward traffic to each other. By design, each of teh access switches connects to at least one distribution switch, typically to two distribution swithces for redundancy. The distribution swithces provide the service of forwarding traffic to other parts of the LAN. Note that most designs use at least two uplinks to two different distribution switches for redundancy.

With a three-tier design you have a core layer which aggregates distribution switches in very large campus LANs, providing very high forwarding rates for the larger volume of traffic dur to the size of the network. The core layer is connected to the distribution switches often with a partial mesh design.

A star topology is a design in which one central device connects to several other, so that if you drew the links out in all directions, the design would look like a star.

A full mesh topology is where each node in the network connects to every other node. $(N(N-1))/2$

A partial mesh is a design that connects a link between some pairs of nodes, but not all.

A hybrid design combines the previous designs into a more complex topology.

Spine-leaf or a Clos network is a network in which every leaf switch must connect to every spine and vice versa. Spine switches cnnaot connect to each other and the same applies for leaf switches. Endpoints connect only to leaf switches. Endpoints can be connections to devices outside the data center, a physical server, a Cisco Unified Computing System (UCS), or the Application Policy Infrastructure Controller (APIC).

WANs span a large geographical area, typically connecting multiple LANs or other types of networks together. These connect offices in different cities or countries. WAN may be Point-to-Point, Hub and Spoke, Mesh, ect.

SOHO is just a slaced down offices network. There are fewer devices but still supports basic network functionality. A SOHO office may be using a star, bus, or some partial mesh design.

On-premise, also known as on-premises, refers to the practice of hosting and managing computing resources within an organization's own physical location, such as data centers or server rooms located on the organization's premises.Typically the org. has full control over both the hardware and software infrastructure. The organization is responsible for everything: maintenance, security, upgrades, and all aspects of IT management. The cost comes from purchasing hardware and ongoing operational costs for maintenance, power, cooling, and space. On=premise may use a star topology or a ring, though a ring is rare in modern networks.

Cloud is increasingly popular for its agility, scalability, and ease of access, suitable for businesses of all sized seeking to reduce infrastructure cost and enhance operational efficiency. Clouds often abstract from physical topologies (overlay) but can inslude any of the above designs depending on the cloud provider's infrastructure and serve offerings. They may use

some hybrid design to allow on-premises to integrate both environments seamlessly.

- **Compare physical interface and cabling types**

  Single-mode fiber is ideal for long-distance transmissions in telecommunications, data centers, and backbone networks where high bandwidth and low attenuation are required. This type of cable uses laser diodes to emit light into a core that is typically 9 microns. Using a single stand of glass fiber can carry data over distances up to tens of kilometers.

  Multi-mode fiber is used for shorter distance transmissions when compared to single-mode. It is used within buildings, campuses, and data centers where high bandwidth is needed. It uses LED or laser diodes to emit light into a larger core (50 or 62.5 microns). It suuports multiple modes of light propagation. Since it uses multiple strands of glass fiber to carry data, it can reliably trasmit data hundreds of meters.

  Copper uses electrical signals to transmit data via copper wires (typically twisted pairs). It is used extensively for Ethernet connections in LANs, connecting computers, switches, routers, and other networking devices within a builing or campus. In Ethernet shared media and point-to-piont devices share the same communication medium, and each device receives all transmissions. Devices on the network must listen for their own MAC address to determine if transmissions is intended for them. Histroically coax was used (10BASEE5 and 10BASEE2) or twisted pair (10BASE-T). Early Ethernet networks used shared media, but it is less common today due to limitations in scalability, collision domains, and performance degradation as network size increases.

  Some relavant Ethernet standards: 802.3, 802.3u, 802.3z, 802.3ab, 802.3an
  copper 100m Ethernet 10 Mbps 10BASE-T
  copper 100m Fast Ethernet 100 Mbps 100BASE-T
  fiber 5000m Gigabit Ethernet 1000 Mbps 1000BASE-LX
  copper 100m Gigabit Ethernet 1000 Mbps 1000BASE-T
  copper 100m 10 Gig Ethernet 10,000 Mbps 10GBASE-T

- **Identify interface and cable issues (collisions, error, mismatch duplex, and/or speed)**

  Collisions occur in Ethernet networks when two devices attempt to transmit data at the same time on a shared medium, such as a network segment. In a collision, the data sent by both devices become corrupted and must be retransmitted, leading to decreased network efficiency and performance issues. High collision rates can indicate network congestion, improper network design, or issues with cabling.

Errors refer to data transmission problems that result in corrupted or lost packets. Errors can occur due to a variety of reasons including electrical interference, faulty cabling or connectors, software bugs, or hardware malfunctions. Errors can lead to data loss, retransmissions, and performance degradation. They are usually indicated by error counters in network device interfaces.

Duplex refers to the ability of a network interface to send and receive data simultaneously. Duplex settings can be either half-duplex or full-duplex. Duplex mismatch can cause degraded network performance, intermittent connectivity issues, and increased error rates.

Speed mismatch occurs when two devices connected on the same network link operate at different transmission speeds. Speed mismatches can lead to connectivity problems, data loss, and inefficient use of network resources. It's important for connected devices to operate at the same speed to ensure smooth communication.

- **Compare TCP to UDP**

  TCP is a connection-oriented protocol. Before data transmission begins, a connection must be established between the sender and receiver (both parties are both senders and receivers). TCP ensures reliable delivery of data by using sequence numbers, acknowledgments, and retransmission of lost packets. It provides reliable, ordered, and error-checked delivery of data packets. It guarantees that data will be delivered intact and in order. If packets are lost or corrupted during transmission, TCP retransmits them until the receiver successfully receives them. TCP has much more overhead work thus its header size is much larger.TCP is typically used for applications that require high reliability and accurate delivery of data, such as web browsing, email, file transfer (FTP), and database transactions. TCP is suited for applications where the integrity of data is paramount and retransmissions are acceptable. TCP implements flow control and congestion control mechanisms to manage the rate of data transmission and avoid network congestion. TCP dynamically adjusts the rate of data transmission based on network conditions to optimize performance and reliability.

  flow control mechanisms:
  Sliding window controler the amount of unacknowledge data that can be in transit between sender and receiver.
  Receiver's advertised window indicates the receiver's available buffer space, influencing the sender's transmission rate.
  TCP ACK mechanism confirms receipt of data segments, guiding the sender's behavior based on acknowledged and unacknowledged data.

congestion avoidance mechanisms:
Slow start initiates the sender's transmission with a conservative window size and exponentially increases it until congestion is deteced or a threshold is reached.
After slow start, linearly increases the congestion window (cwnd) to balance efficient data transmission with congestion prevention.
TCP vegas uses delay measurements to detect congestion, adjusting the sending rate to maintain low latency.
Explicit Congestion Notification (ECN) allows routers to mark packets instead of dropping them to notify TCP about congestion, enabling proactive rate adjustment.

congestion response mechanisms:
Fast retransmit reissues a segment up receiving duplicate acknowledgments to expedite recovery without waiting for timeout.
Fast recovery temporarily reduces the congestion window upon packet loss, facilitating faster recovery to maintain efficient data flow.
Selective Acknowledgment (SACK) allows the receiver to acknowledge out-of-order segments, improving TCP performance in environments with high packet loss. Dynamic Congestion Window Scaling adjusts the congestion window dynamically based on network conditions to optimize throughput and minimize congestion.

additional considerations:
Timeout and Retransmission: TCP implements timers to detect lost packets and retransmits them if necessary after a timeout period.
Window Scaling Options: Supports larger window sizes beyond the traditional 16-bit limit to enhance performance on high-bandwidth networks.
Early Retransmit: Provides an alternative to traditional retransmission for certain network conditions, improving responsiveness.
Congestion Control Algorithms: Besides Reno, TCP also includes NewReno, CUBIC, and other variants optimized for specific network characteristics and performance metrics.

UDP is a connectionless protocol. It does not establish a connection before sending data and does not guarantee delivery or order of packets. UDP is more lightweight and efficient for applications that can tolerate some packet loss, such as real-time multimedia streaming or online gaming. It does not provide reliability or guaranteed delivery. It does not retransmit lost packets or ensure that they are received in order. Applications using UDP must handle these aspects themselves if needed. Since there is less overhead work it is more efficient for transmitting data where reliability and ordering are less critical. It is best suited for applications where real-time and low-latency transmission is more important than guaranteed

delivery, such as video streaming, voice over IP (VoIP), online gaming, DNS (Domain Name System), and IoT applications.

- **Configure and verify IPv4 addressing and subnetting**

  configure: Router(config-if)# ip address <ip_address> <subnet_mask>

  verify: Router# show ip interface brief

- **Describe private IPv4 addressing**

  Unstructuered Addressing (pre-1981)
  In the earliest days of networking and the ARPANET, there was no formalized IP addressing scheme. Networks were small and interconnected in a limited manner, often using ad-hoc addressing methods or relying on manual configuration.

  Early Structured Addressing (Pre-Classful Addressing 1981 - early 1990s)
  In the early stages of IPv4 development and the initial years of networking, there was a gradual move towards more structured approaches to IP addressing. This period predates the formal introduction of classful addressing (which started in 1981) and can be summarized by: ad-hoc addressing, manual configuration, early standardization efforts.

  Classful Addressing (1981 - early 1990s)
  Classful addressing was introduced around 1981 as a standardized approach to IPv4 addressing. It divided the IPv4 address space into five distinct classes, denoted by the high-order bits of the IP address. EAch class had a specific range of addresses and a default subnet mask, intended to accommodate networks of different sizes:// Class A (0.0.0.0 to 127.255.255.255) /8: used for large networks with a small number of high-capacity hosts
  Class B (128.0.0.0 to 191.255.255.255) /16: used for medium-sized networks
  Class C (192.0.0.0 to 223.255.255.255) /24: used for small networks with a large number of hosts
  Class D (224.0.0.0 to 239.255.255.255): used for multicast groups
  Class E (240.0.0.0 to 255.255.255.255): intended for research and development purposes and are not routable on the public internet

  Transition to Classless Inter-Domain Routing (CIDR) (early 1990s)
  Variable Length Subnet Masks (VLSM) allows for the use of subnet masks of varying lengths, not limited to the fixed boundaries. Network administrators can subnet a network into smaller subnets of different sizes according to their specific needs. This allows for more efficient use of IP address space by allocating addresses in smaller, more manageable blocks.
  Address aggregation also known as supernetting or prefix aggregation reduces the size of routing tables by combining multiple contiguous IP address blocks into a single route advertisement. This help minimize the

number of routing table entries in routers across the internet, improving routing efficiency and reducing the overhead associated with routing updates.

IPv4 private addressing refers to a range of IP addresses designed for use within private networks that are not routable on the public internet. These address are specified in RFC 1918 and are commonly used in home, office, and enterprise networks for internal communication.
10.0.0.0/8
172.16.0.0/12
192.168.0.0/16

The public IP address range refers to the IP addresses that are globally routable on the internet allowing devices to communicate directly with each other across different networks. Unlike private IP addresses, public IP addresses are unique and assigned by IANA to organizations, ISPs, or other entities that require connectivity to the internet. These addresses are used for devices and services that need to be accessible from the public internet. The public IP address space in IPv4 is managed through various RIRs around the world. The IANA has allocated serveral ranges of IPv4 addresses to these RIRs, who then assign them to ISPs and organization based on regional needs.
1.0.0.0 to 126.255.255.255 (excluding 10.0.0.0/8)
128.0.0.0 to 191.255.255.255 (excluding 172.16.0.0/12)
192.0.0.0 to 223.255.255.255 (excluding 192.168.0.0/16)

- **Configure and verify IPv6 address types**

  configure: Router(config-if)# ipv6 address <IPv6_address>/<prefix_length>

  verify: Router# show ipv6 interface brief

- **Describe IPv6 address types**

  Unicast is used for one to one communication.
  Global unicast addresses are equivalent to public IPv4 addresses and are routable on the IPv6 internet. They consist of a 48-bit global routing prefix, a 16-bit subnet ID for hierarchical addressing, and a 64-bit interface ID to identify specific devices within a subnet. The global routing prefixc is assigned by internet registeries (ARIN, RIPDE NCC, APNIC) to ISPs and large organizations.
  FC00::/7 Unique local addresses (ULA) are equivalent to IPv4 private addresses and are used for local communication within a site or organization. They are not routable on the global IPv6 internet.
  FE80::/10 Link-local addresses are used for communication within the same network segment. They are not routable beyond the immediate link.

Anycast is used for one-to-nearest communication, typically for load balancing. These are common in DNS servers or Content Delivery Networks. (CDN)

FF00::/8 Multicast is used for one-to-many communication, distributing data efficiently to multiple recipients simultaneously.

Modified EUI-64 is derived from a modified MAC address, used for automatic address configuration. Typically this is used for global unicast and link-local addresses. Insert FF in the middle of the address then invert the 7th bit and finally append the interface ID.

- **Verify IP parameters for Client OS**

  ipconfig /all /release /renew

  ifconfig -a

- **Describe wireless principles**
- **Explain virtualization fundamentals (server virtualization, containers, and VRFs)**
- **Describe switching concepts**

Network Access

- **Configure and verify VLANs (normal range 0-1005) spanning multiple switches**

  1. create vlans, 2. port operation, 3. apply vlans
  Switch(config)# vlan <vlan-id>
  Switch(config-if)# switchport mode access
  Switch(config-if)# switchport access vlan <vlan-id>

  verify: defined/active VLANs, VTP, access ports(data and voice)
  Switch# show vlan [brief]
  Switch# show interface <interface>[brief]
  Switch# show interfaces status
  Switch# show run

- **Configure and verify interswitch connectivity**

  1. operational mode, 2. allowed vlans, 3. encapsulation standard, 4. native vlan
  Switch(config-if)# switchport mode trunk
  Switch(config-if)# switchport trunk allowed vlan <10,20,30>
  Switch(config-if)# switchport tunk encapsulation <dot1q/isl>
  Switch(config-if)# switchport trunk native vlan <vlan-id>

verify: VLANs, native VLAN, VTP, DTP, operational mode
Switch# show interface trunk
Switch# show interfaces switchport
Switch# show interface <interface>

- **Configure and verify L2 discovery protocols (CDP/LLDP)**

  CDP: multicast address 0100.0ccc.cccc
  Switch(config-if)# cdp enable
  Switch(config)# cdp run
  Switch(config)# cdp timer seconds
  Switch(config)# cdp holdtime seconds

  verify↓
  Switch# show cdp
  Switch# show cdp traffic
  Switch# show cdp <interface>
  Switch# show cdp neighbors
  Switch# show cdp neighbors detail
  Switch# show cdp entry name

  LLDP(802.1AB): multicast address 0180.c200.000e
  Switch(config-if)# lldp enable
  Switch(config)# lldp run
  Switch(config-if)# lldp transmit
  Switch(config-if)# lldp receive
  Switch(config)# lldp timer seconds
  Switch(config)# lldp holdtime seconds

- verify↓
  Switch# show lldp
  Switch# show lldp traffic
  Switch# show lldp interface
  Switch# show lldp neighbors
  Switch# show lldp neighbors detail
  Switch# show lldp entry name

- **Configure and verify (L2/L3) EtherChannel (LACP)**

  Manual Configuration↓
  Switch(config-if-range)# channel-group 1 mode on
  Switch(config)# port-channel load-balance method

  Dynamic Congifuration↓
  Switch(config-if-range)# channel-group 1 mode <desirable/auto>(PAgP)
  Switch(config-if-range)# channel-group 1 mode <active/passive>(LACP)

  verify: speed, duplex, operational mode, VLANs, STP
  Switch# show etherchannel summary
  Switch# show interfaces status

L3 Etherchannel configuration is similar to L2 configuration:
1. each physical port needs to be a routed port (no switchport command)
2. also the actuall channel needs to be a routed port with an IP and mask

- **Interpret basic operations of Rapid PVST+ Spanning Tree Protocol**

  Root port, root bridge (primary/secondary), and other port names

  port states and roles

  PortFast

  Root guard, loop guard, BPDU filter, and BPDU guard

- **Describe Cisco Wireless Architectures and AP modes**
- **Describe physical infrastructure connections of WLAN components (AP, WLC, access/trunk ports, and LAG)**
- **Describe network device management access (Telnet, SSH, HTTP, HTTPS, console, TACACS+/RADIUS, and cloud managed)**
- **Interpret the wireless LAN GUI configuration for client connectivity, such as WLAN creation, security settings, QoS profiles, and advanced settings**

IP Connectivity

- **Interpret the compnents of routing table**
- **Determine how a router makes a forwarding decision by default**
- **Configure and verify IPv4 and IPv6 static routing**
  NOTE: using the link-local address as the next hop address is ambigious you must include the outgoing-interface beforehand
  NOTE: IOS allows you to configure the ipv6 route command using only the outgoing-interface parameter, without listing a next-hop address. The router will accept the command; however, if that outgoing interface happens to be an Ethernet interface, the router cannot successfully forward IPv6 packets using the route.

  Default routes are used when a packet's destination address does not match any routes, but you do not want the router to discard a packet that it otherwise would.
  Router(config)# ip route 0.0.0.0 0.0.0.0 <interface>
  Router(config)# ipv6 router ::/0 <interface>

Network routes define a route to an entire subnet.
Router(config)# ip route 172.16.2.0 255.255.255.0 s0/0/0
Router(config)# ip route 172.16.3.0 255.255.255.0 172.16.5.3
Router(config)# ipv6 route 2001:db8:1111:2::/64 s0/0/0

Host routes match a single IP. An engineer may want to route most packets to some subnet through one route but packets destined for a specific host through another route.
Router(config)# ip route 10.1.1.9 255.255.255.255 10.9.9.9
Router(config)# ipv6 router 2001:db8:1111:2::22/128 s0/0/0 fe80::ff:fe00:2

Floating static routes are used when when a primary route fails.
Router(config)# ip route 172.16.2.0 255.255.255.0 172.16.5.3 130
Router(config)# ipv6 route 2001:db8:1::/64 2001:db8:2::1 200

- **Configure and verify single area OSPFv2**

Direct↓
Router(config)# ospf process <1-65535>
Router(config-router)# router-id X.X.X.X (OPTIONAL) Router(config-router)# network <IP_address><wildcard>area <area_id>

Indirect↓
Router(config-if)# ip ospf process <1-65535><area_id>
Router(config-if)# ip ospf network [type] (OPTIONAL)

Passive interfaces do not send Hellos but information about the connected subnet will still be advertised elsewhere.
Router(config-router)# passive-interface <interface>

OSPF default routes work like normal default routes, but can be advertised via OSPF
Router(config-router)# default-information originate

OSPF cost = reference / bandwidth
Router(config-if)# ip ospf cost cost
Router(config-router)# auto-cost reference
Router(config-if)# speed 100
Router(config-if)# bandwidth 10000
NOTE: the bandwidth command is used in scenarios where the actual bandwidth of the interface differs from the default assumption made by the routing protocol.
Router(config-if)# clock
NOTE: the clock command is normally configured on the DCE end of a serial link to provide clocking for the line.

OSPF route summarization involves aggregating multiple contiguous network addresses into a single summary route advert.
NOTE: Route summarization should be done at the netwrok boundary to

avoid potential routing issues. Also, make sure that summarized routes cover all the individual routes being summarized.
Router(config-router)# area 0 range <IP_address><subnet_mask>

Routing Black Holes:
A routing black hole occurs when a router receives traffic for a destination network but does not have a valid route to forward that traffic. This can happen due to various reasons, including misconfigured routes, unreachable next hops, or route summarization that omits specific routes. When a router encounters a routing black hole, it frops the packets, resulting in loss of connectivity to the affected destination.

Suboptimal Routing Decisions:
Suboptimal routing decisions refer to situations where routers select paths that are not the most efficient or optimal routes to reach a destination. This can occur due to factos such as unequal cost load balancing, asymmetric routing, or inefficient routing protocols. Suboptimal routing decisions can lead to increased latency, congestion, and subpar network performance.

verify: MTU, areas, network types, timers, neighbor states/roles, reference bandwidth, authentication
Router(config)# show run (config)
Router(config)# show ip protocols (config)

Router(config)# show ip ospf interface (enabled interfaces)
Router(config)# show ip ospf interface <interface>(enabled interfaces)
Router(config)# show ip ospf interface brief (enabled interfaces)

Router(config)# show ip ospf neighbor (neighbors)
Router(config)# show ip ospf neighbor <interface>(neighbors)

Router(config)# show ip ospf database (lsdb)

Router(config)# show ip ospf rib (rib)

Router(config)# show ip route (routes)
Router(config)# show ip route ospf (routes)
Router(config)# show ip route subnet mask (routes)
Router(config)# show ip route — section subnet (route)

- **Describe the purpose, functions, and concepts of first hop redundancy protocols**

IP Services

- **Configure and verify inside source NAT using static and pools**

  Static NAT
  Router(config-if)# ip nat inside
  Router(config-if)# ip nat outside
  Router(config)# ip nat inside source static <insidelocal><outsidelocal>

  Dynamic NAT
  Router(config)# ip nat pool mypool 203.0.113.20 203.0.113.30 netmask 255.255.255.0
  Router(config)# access-list 1 permit 192.168.1.0 0.0.0.255
  Router(config)# ip nat inside source list 1 pool mypool

  verify: interfaces, address ranges, acl, hit/misses, routes to destinations↓
  R1# show access-lists
  R1# show ip nat translations [verbose]
  R1# show ip nat statistics
  R1# show ip route
  R1# show run
  R1# show interfaces <interface>

- **Configure and verify NTP operating in a client and server mode**

  NOTE: the time-range command can be used to set maintenance windows

  software clock↓
  Router# clock set hh:mm:ss month day year

  hardware clock↓
  Router# calendar set hh:mm:ss date dd month

  timezones↓
  Router(config)# clock timezone name hours-offset [minutes-offset]

  daylight savings↓
  Router(config)# clock summer-time recurring name start end [offset]

  NTP uses UTC time standard to sync network devices
  R1(config)# ntp update-calendar
  R1(config)# interface loopback0
  R1(config-if)# ip address 10.1.1.1 255.255.255.255
  R1(config)# ntp source loopback 0
  R1(config)# ntp master <1-15>
  R2(config)# ntp server 10.1.1.1 [prefer]
  R2(config)# ntp peer 10.0.23.3
  R3(config)# ntp peer 10.0.23.2
  R3(config)# ntp authenticate

R3(config)# ntp authentication-key key-number md5 key
R3(config)# ntp trusted-key key-number
R3(config)# ntp server 10.1.1.1 key key-number

verify ↓
Router# show ntp associations
(* sys.peer, # selected, + candidate, - outlyer, x falseticker, configured)
Router# show ntp status

- **Explain the role of DHCP and DNS within the network**
- **Explain the function of SNMP in network operations**
- **Describe the use of syslog features including facilities and levels**
- **Configure and verify DHCP client and relay**

  DHCP relay is used to help move DHCP messages in/out of a segment
  Router(config-if)# ip helper-address 192.168.1.100

  show commands ↓
  R1# show ip dhcp binding
  R1# show ip dhcp pool
  R1# show ip dhcp server statistics
  R1# show ip dhcp relay
  ipconfig/ifconfig (verify clients DHCP information)

- **Explain the forwarding per-hop behavior (PHB) for QoS, such as classification, marking, queuing, congestion, policing, and shaping**
- **Configure network devices for remote access using SSH**

  1. enable SSH 2. create user accounts 3. set parameters 4. access control (OPTIONAL)
  Router(config)# crypto key generate rsa
  Router(config)# username <username>priviledge 15 secret <pass>
  Router(config)# ip ssh version 2
  Router(config)# ip ssh time-out 120
  Router(config)# ip ssh authentication-retries 3

  verify by attempting a remote login
  Router# show run — include ssh
  Router# show ip ssh

- **Describe the capabilities and functions of TFTP/FTP in the network**

Security Fundamentals

- **Define key security concepts (threats, vulnerabilities, exploits, and mitigation techniques)**
- **Describe security program elements (user awareness, training, and physical access control)**
- **Configure and verify device access control using local passwords**

  configure↓ Router(config)# service password-encryption

  Router(config)# enable secret <password>

  Router(config)# line console 0
  Router(config-line)# password <password>
  Router(config-line)# login

  Router(config)# line vty 0 15
  Router(config-line)# password <password>
  Router(config-line)# login

  verify↓ Router# show run — include enable secret Router# show run — include line console 0 Router# show run — include line vty 0 15

- **Describe security password policies elements, such as management, complexity, and password alternatives (multifactor authentication, certificates, and biometrics)**
- **Describe IPsec remote access and site-to-site VPNs**
- **Configure and verify access control lists**

  Standard ACLs match source IP addresses only.
  R1(config)# access-list 1 deny [host] 192.168.1.233 [log]

  Extended ACLs have more matching parameters.
  R1(config)# access-list 101 permit tcp 172.16.1.0 0.0.0.255 172.16.3.0 0.0.0.255 eq 21

  Named ACLs have a subconfiguration mode with sequence numbers
  Router(config)# ip access-list extended barney
  Router(config-ext-nacl)# permit tcp host 10.1.1.2 eq www any
  Router(config-ext-nacl)# interface serial1
  Router(config-if)# ip access-gorup barney out

  verify↓
  Place standard ACLs as close to the destination as possible.
  Place extended ACLs as close to the source as possible.
  Disable an ACL before altering the ACEs.

some arithmetic ↓
subnet = host address &(binary and) mask
wildcard mask = limited broadcast address - subnet mask
subnet(highend) = subnet(lowend) + wildcard
0d → 0b ↓
1. continue to divide by 2
2. each division set aside the remainder
3. once the quotient is 0, merge the remainders in reverse order
0x → 0d ↓
1. separate the digits and convert them to decimal digits
2. multiply each digit by the respective power of 16
3. add them
0d → 0x ↓
1. identify the largest power of 16 less than the decimal number
2. continue to divide the remainders by the largest power of 16 setting the quotient aside
3. merge quotients and convert the digits individually

- **Configure and verify Layer 2 security features (DHCP snooping, dynamic ARP inspection, and port security)**

  DHCP snooping is used to prevent spurious DHCP servers
  Switch(config)# ip dhcp snooping
  Switch(config)# ip dhcp snooping vlan <vlan_id>
  Switch(config-if)# ip dhcp snooping trust verify↓
  Switch# show ip dhcp snooping
  Switch# show ip dhcp snooping binding

  Dynamic ARP inspection is used prevent ARP spoofing/poisoning
  Switch(config)# ip arp inspection
  Switch(config)# ip arp inspection vlan <vlan_id>
  Switch(config-if)# ip arp inspection trust verify↓ Switch# show ip arp inspection
  Switch# show ip arp inspection statistics

  Port security is used to keep unauthorized devices from accessing the network.
  Switch(config-if)# switchport port-security
  Switch(config-if)# switchport port-security maximum <value>
  Switch(config-if)# switchport port-security violation shutdown — restrict — protect
  verify↓ Switch# show port-security interface <interface>
  Switch# show port-security [address]

- **Compare authentication, authorization, and accounting concepts**
- **Describe wireless security protocols (WPA, WPA2, and WPA3)**

- **Configure and verify WLAN within the GUI using WPA2 PSK**

  1. Access the GUI
  2. Login
  3. Navigate to WLAN Settings
  4. Create a New WLAN Profile
  5. Configure WLAN Settings: SSID, security settings, pre-shared key, encryption, etc.
  6. Apply and Save Changes
  7. Verify WLAN Configuration
  8. Test Connectivity

Automation and Programmability

- **Explain how automation impacts network management**

  Automation allows a network to be changed all at once. This minimizes the amount of times humans have to interface with the network. This provides consistency which will decrease the amount of misconfigured devices and thus improve security. Templates and policies can be implemented up-front to comply with organizational standards and industry best practices. Automated tools can be implemented for monitoring and maintenance these tools continuously track network performance metrics and detect anomalies by integrating ML. ML can also be used to imrpove the turn-around time once certain metrics are observed. This saves time, money, and improves the QoE for users of the network. Automation simplifies audit processes by generating comprehensive reports on network configurations, access controls, and compliance with regulatory requirements, facilitating easier compliance management. Since automation reduces the need for manual, repetitive task network administration time can be properly allocated to the most important strategic initiatives.

- **Compare traditional networks with controller-based networking**

  In a traditional network you have a hierarchical structure, control is distributed evenly amoungst the network devices and each devices operates independently and makes its own desisions. Management is done manually with little automation. These networks do not scale well and optimizations are managed via static routing and other manual adjustments. These traditional networks are much less flexible when new inovations become popular because there are many constraints that are not in alignment. Fault tolerance is typically achived via hardware redundacy and protocols which can be costly the more you scale. In opposition, SDN architectures

are flattened and the control plane functions are decoupled from the data plane functions and placed in a central location. Management is done via the controller and network devices and their data can be read/written automatically via APIs. This type of network architeture scales better and is more fault tolerant because you can dynamically allocate resources and adjust network behavior through policies.

- **Describe controller-based, software defined architecture (overlay, underlay, and fabric)**
- **Explain AI (generative and predictive) and machine learning in network operations**

  Generative AI involves systems capable of creating new content, data, or solutions based on patterns learned from existing data. Generative AI can optimize network topologies based on parameters like performance requirements, scalability, and redundancy. It can automate the configuration process for devices and services. It can leverage NLP for natrual language quireies, interpretting logs, and support tickets. This enables automated responses, troubleshooting advice, and root cause analysis.

  Predictive AI focuses on forecasting future outcomes based on historical data and patterns. In networking operations, predictive AI enhances decision-making and proactice management. Algorithms analyze network equipment performance data (latency, packet loss, and hardware metrics) to predict potential failures or degradation. This allows for proactive maintenance and minimizes downtime. Models can detect anomalies in network traffic patterns that may indicate security threats, such as DDOS attacks, malware infiltration, or unauthorized access attempts. By analyzing historical attack data and network vulnerabilities, predictive AI can forecast potential security threats and recommend preemptive measures.

  Limitations of generative AI: complexity and accuracy, interpretability, resouce intensiveness, adaptability to dynamic environments, ethical and security concerns
  Generative AI models may struggle with the complexity of network configurations and interactions, especially in large-scale and heterogeneous networks.
  Accuracy in generating optimal network designs or configurations requires extensive training data and robust algorithms.
  Understanding and interpreting outputs from generative AI models can be challenging. It may be difficult to explain why a specific network design or configuration was chosen by the AI.
  Training and running generative AI models often require significant computational resources and time. This can be impractical for real-time or near-real-time network management tasks.
  Generative AI models may struggle to adapt quickly to dynamic changes

in network conditions, such as fluctuating traffic patterns or new technology integrations.
AI-generated network configurations or optimizations may introduce vulnerabilities if not thoroughly tested or validated. Security implications of AI-generated decisions must be carefully assessed. (cat and mouse game)

Limitations of predictive AI data quality and availability, prediction uncertainty, complexity of network interactions, scalability and generalization, human expertise and validation: Predictive AI models heavily rely on historical data for accurate forecasting. Poor data quality or insufficient historical data can lead to inaccurate predictions.
Network environments may have varying data quality and consistency, affecting the reliability of predictions.
Predictive AI models may struggle with uncertainty and variability in network behaviors, especially during unforeseen events or unusual network conditions.
Overfitting to historical data or underfitting to new patterns can lead to inaccurate predictions.
Predictive AI models may oversimplify network interactions and dependencies, leading to incomplete or inaccurate predictions of performance or security risks.
Capturing the full complexity of network dynamics, including interactions between devices, protocols, and applications, remains a challenge.
Scaling predictive AI models to large and diverse network environments while maintaining accuracy and performance can be difficult.
Generalizing predictive models across different types of networks or organizational contexts may require extensive customization and validation.
Despite advancements in AI, human expertise and validation are essential for interpreting predictions, validating recommendations, and making informed decisions in network operations.

Mitigating Strategies: data governance, hybrid approches, continuous learning, robust testing and validation, ethical considerations

- **Describe characteristics of REST-based APIs (authentication types, CRUD, HTTP verbs, and data encoding)**

  Representational State Transfer: stateless, client-server architecture, uniform interface, resource-based, state transfer, cacheability, layered system, code-on-demand (OPTIONAL)

  Authentication in REST-based APIs can be implemented in various ways:
  HTTP Basic Authentication: This involves sending the username and password with each request using the 'authorization' header. It's simple but less secure because credentials are sent in plaintext.
  HTTP Digest Authentication: Similar to Basic Authentication but hashes the credentials to improve security.

Token-based Authentication (JWT): Tokens are issued after initial authentication and are sent with each subsequent request. JWT (JSON Web Token) is a popular choice due to its compact size and ability to contain user information.
OAuth: OAuth 2.0 is widely used for delegated authorization, allowing third-party applications to access resources on behalf of a user without exposing their credentials.
API Keys: A simple form of authentication where an API key (a unique identifier) is included in the request header or URL query parameters.

HTTP verbs nicely map onto CRUD operations
GET: retrieves data from the server
POST: submits data to be processed to the server
PUT: updates data on the server (replacing existing data)
PATCH: updates data on the server (partially modifying existing data)
DELETE: removes data from the server

Data Encoding: converting structured data or objects into a format that can be easily stored, transmitted, or reconstructed later. (saving data to disk, sending it over a network, sharing is between systems)
JavaScript Objext Notation (JSON) is a lightweight data interchange format that is easy for humans to read and write, and easy for machines to parse and generate.
eXtensible Markup Language (XML) provides a hierarchical structure and is used for structued data representation. It allows defining custom data types and structures.
Binary Formats like Protocol Buffers (protobuf), MessagePack, and BSON are optimized for compactness and efficient serialization/deserialization.

- **Recognize the capabilities of configuration management mechanisms, such as Ansible and Terraform**

  Ansible is an open-source automation tool used for configuration management application deployment, task automation, and orchestration.
  Ansible operates over SSH, so there is no need to install an agent on devices. Ansible uses YAML (serializatoin language) for writing playbooks, which define tasks and configurations to be executed on remote systems. Ansible itself interprets the YAML playbook and executes task using modules and plugins. YAML itself does not contain executable code, its structured nature and readability make it a versatile choice for defining configurations, workflows, and data representations that can be interpreted and acted upon by other software systems or automation tools. Ansible is idempotent so if the system is already in the desired state, running the playbook again will not cause any changes. It is extensible so modules can be custom-written to support various tasks and systems.

  Terraform is an open-source infrastructure as code (IaC) tool used for defining and provisioning infrastructure across multiple cloud providers.

Infrastructure is defined in configuration files using HashiCorp Configuration Language (HCL) or JSON. Terraform builds a dependency graph of resources and manages them accordingly. It supports provisioning resources across various cloud providers like AWS, Azure, Google Cloud, etc. Terraform keeps track of the state of the infrastructure allowing for updates and modifications without downtime.

- **Recognize components of JSON-encoded data**

  An object in JSON is enclosed in curly braces '' and consists of key-value pairs. Keys must be strings, and values can be any valid JSON data type (string, number, object, array, boolean, null)

  strings will be a sequence of characters enclosed in double quotes '"'"'

  JSON does not differentiate between integers and floating-point numbers in terms of syntax; it treats all numbers as numeric values.

  booleans are either 'true' or 'false'

  arrays are on ordered collection of values, enclosed in square brackets '[]'. Arrays can contain values of any JSON data type, including other arrays (nested arrays).

  Null represents an empty or null value.

  NOTE: JSON itself is not a programming language and does not have the extensive type system that programming languages do (such as strongly-typed or dynamically-typed systems). Instead, JSON defines a set of data structures and rules for representing data in a way that is both human-readable and machine-parseable. This simplicity and flexibility make JSON widely used for data interchange between systems, especially in web dev and APIs.